

index.js > ...

```
1  const Autor = require('./models/Autor');
2  const Livros = require('./models/Livros');
3
4  const criandoAutor = {
5    nome: 'Fernando Pessoa',
6    biografia: 'Nasceu em 1888, em Lisboa.'
7  };
8
9  Autor.atualizarAutor(5, criandoAutor);
10
```

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Autor atualizado: 1

```
index.js
1  const Autor = require('./models/Autor');
2  const Livros = require('./models/Livros');
3
4  const criandoAutor = {
5    nome: 'Fernando Pessoa',
6    biografia: 'Nasceu em 1888, em Lisboa.'
7  };
8
9  const criandoLivro = {
10   titulo: 'Mensagem',
11   genero: 'Poesia',
12   autor_id: 5
13 }
14
15 Livros.atualizarLivros(3, criandoLivro)
```

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Livro atualizado: 1

➤

```
index.js > ...  
1  const Autor = require('./models/Autor');  
2  const Livros = require('./models/Livros');  
3  
4  const criandoAutor = {  
5    nome: 'Carlos Drummond',  
6    biografia: 'Nasceu em 1902, em Itabira'  
7  };  
8  
9  Autor.criarAutor(criandoAutor);
```

```
sqlite does not support inserting default values. Set the `useNull  
AsDefault` flag to hide this warning. (see docs https://knexjs.org  
/guide/query-builder.html#insert).  
sqlite does not support inserting default values. Set the `useNull  
AsDefault` flag to hide this warning. (see docs https://knexjs.org  
/guide/query-builder.html#insert).  
Hint: hit control+c anytime to enter REPL.  
Autor criado com ID: 4  
❯
```

index.js

```
1 const Autor = require('./models/Autor');
2 const Livros = require('./models/Livros');
3
4 const criandoAutor = {
5   nome: 'Carlos Drummond',
6   biografia: 'Nasceu em 1902, em Itabira'
7 };
8
9 const criandoLivro = {
10   titulo: 'A Rosa do Povo',
11   genero: 'Poesia',
12   autor_id: 5
13 }
14
15 Livros.criarLivros(criandoLivro);
```

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Livro criado com ID: 3

➤

JS index.js > ...

```
1  const Autor = require('./models/Autor');
2  const Livros = require('./models/Livros');
3
4  const criandoAutor = {
5    nome: 'Carlos Drummond',
6    biografia: 'Nasceu em 1902, em Itabira'
7  };
8
9  Autor.excluirAutor(4)
```

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Autor excluído: 1

✱

JS index.js

```
1  const Autor = require('../models/Autor');
2  const Livros = require('../models/Livros');
3
4  const criandoAutor = {
5    nome: 'Fernando Pessoa',
6    biografia: 'Nasceu em 1888, em Lisboa.'
7  };
8
9  const criandoLivro = {
10    titulo: 'Mensagem',
11    genero: 'Poesia',
12    autor_id: 5
13  }
14
15  Livros.excluirLivros(3)
```

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Livro excluído: 1

> []

index.js

```
1  const Autor = require('./models/Autor');
2  const Livros = require('./models/Livros');
3
4  const criandoAutor = {
5    nome: 'Fernando Pessoa',
6    biografia: 'Nasceu em 1888, em Lisboa.'
7  };
8
9  Autor.obterAutor(5)
10
11 |
```

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNullAsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

```
Autor encontrado: {
  id: 5,
  nome: 'Fernando Pessoa',
  biografia: 'Nasceu em 1888, em Lisboa.'
}
> []
```

index.js

```
1  const Autor = require('./models/Autor');
2  const Livros = require('./models/Livros');
3
4  const criandoAutor = {
5    nome: 'Fernando Pessoa',
6    biografia: 'Nasceu em 1888, em Lisboa.'
7  };
8
9  const criandoLivro = {
10    titulo: 'Mensagem',
11    genero: 'Poesia',
12    autor_id: 5
13  }
14
15  Livros.obterLivros (3);
```

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

sqlite does not support inserting default values. Set the `useNull AsDefault` flag to hide this warning. (see docs <https://knexjs.org/guide/query-builder.html#insert>).

Hint: hit control+c anytime to enter REPL.

Livro encontrado: { id: 3, titulo: 'Mensagem', genero: 'Poesia', autor_id: 5 }

> []

knexfile.js > <unknown> > development > ...

```
1  module.exports = {  
2    development: {  
3      client: 'sqlite3',  
4      connection: {  
5        filename: './SCA.db'  
6      },  
7      useNullAsDefault: true  
8    }  
9  };
```

migrations >  20230619164417_create_tables.js > f up > f down > f up > f down > f up > ...

```
1  exports.up = function(knex) {
2
3    return knex.schema
4    .createTable('Autor', function(table) {
5      table.increments('id');
6      table.string('nome').nullable().unique();
7      table.string('biografia').nullable();
8    })
9    .createTable('Livros', function(table) {
10     table.increments('id');
11     table.string('titulo').nullable().unique();
12     table.string('genero').nullable();
13     table.integer('autor_id').unsigned();
14     table.foreign('autor_id').references('Autor.id')
15   });
16 };
17
18 exports.down = function(knex) {
19
20   return knex.schema
21   .dropTableIfExists('Autor')
22   .dropTableIfExists('Livros');
23
24 };
25
```

models > Autor.js > Autor > listarAutores > ...

```

1  const knex = require('knex');
2  const database = knex({
3    client: 'sqlite3',
4    connection: {
5      filename: 'SCA.db'
6    }
7  });
8
9  class Autor {
10   static async criarAutor(autor) {
11     try {
12       const [id] = await database('Autor').insert(autor);
13       console.log('Autor criado com ID:', id);
14     } catch (error) {
15       console.error('Erro ao criar autor:', error);
16     } finally {
17       database.destroy();
18     }
19   }
20
21   static async atualizarAutor(id, autor) {
22     try {
23       const result = await database('Autor').where({ id }).update(autor);
24       console.log('Autor atualizado:', result);
25     } catch (error) {
26       console.error('Erro ao atualizar autor:', error);
27     } finally {
28       database.destroy();
29     }
30   }
31
32   static async obterAutor(id) {
33     try {
34       const autor = await database('Autor').where({ id }).first();
35       console.log('Autor encontrado:', autor);
36     } catch (error) {
37       console.error('Erro ao obter autor:', error);
38     } finally {
39       database.destroy();
40     }
41   }
42
43   static async excluirAutor(id) {
44     try {
45       const result = await database('Autor').where({ id }).del();
46       console.log('Autor excluído:', result);
47     } catch (error) {
48       console.error('Erro ao excluir autor:', error);
49     } finally {
50       database.destroy();
51   }

```

```
52     }  
53  
54     static async listarAutores() {  
55         try {  
56             const result = await database('Autor').getAll();  
57             console.log('Todos os autores:', result);  
58         } catch (error) {  
59             console.error('Erro ao listar autores:', error);  
60         } finally {  
61             database.destroy();  
62         }  
63     }  
64 }  
65  
66 module.exports = Autor;
```

models > Livros.js > ...

```
1  const knex = require('knex');
2  const database = knex({
3    client: 'sqlite3',
4    connection: {
5      filename: 'SCA.db'
6    }
7  });
8
9  class Livros {
10   static async criarLivros(livros) {
11     try {
12       const [id] = await database('Livros').insert(livros);
13       console.log('Livro criado com ID:', id);
14     } catch (error) {
15       console.error('Erro ao criar livro:', error);
16     } finally {
17       database.destroy();
18     }
19   }
20
21   static async atualizarLivros(id, livros) {
22     try {
23       const result = await database('Livros').where({ id }).update(livros);
24       console.log('Livro atualizado:', result);
25     } catch (error) {
26       console.error('Erro ao atualizar livro:', error);
27     } finally {
28       database.destroy();
29     }
30   }
31
32   static async obterLivros(id) {
33     try {
34       const livros = await database('livros').where({ id }).first();
35       console.log('Livro encontrado:', livros);
36     } catch (error) {
37       console.error('Erro ao obter livro:', error);
38     } finally {
39       database.destroy();
40     }
41   }
42
43   static async excluirLivros(id) {
44     try {
45       const result = await database('Livros').where({ id }).del();
46       console.log('Livro excluído:', result);
47     } catch (error) {
48       console.error('Erro ao excluir livro:', error);
49     } finally {
50       database.destroy();
51   }
```

```
52     }
53
54     static async listarLivros() {
55         try {
56             const result = await database('Livros').getAll();
57             console.log('Todos os Livros:', result);
58         } catch (error) {
59             console.error('Erro ao listar livros:', error);
60         } finally {
61             database.destroy();
62         }
63     }
64 }
65
66 module.exports = Livros;
67
```

package.json > ...

```
1 {  
2   "name": "nodejs",  
3   "version": "1.0.0",  
4   "main": "index.js",  
5   "scripts": {  
6     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
7   },  
8   "keywords": [],  
9   "author": "",  
10  "license": "ISC",  
11  "dependencies": {  
12    "@types/node": "^18.0.6",  
13    "knex": "^2.4.2",  
14    "node-fetch": "^3.2.6",  
15    "sqlite3": "^5.1.6"  
16  },  
17  "description": ""  
18 }  
19
```

index.js × knexfile.js × 20230619164417_create... × 01_seed_data.js × 02_seed_data.js × Autor.js × +

seeds > 01_seed_data.js > seed > seed > ...

```
1  exports.seed = function(knex) {
2    return knex('Autor')
3      .del()
4      .then(function() {
5        return knex('Autor').insert([
6          { nome: 'Autor 1', biografia: 'Nasceu em 1981, em Salvador.' },
7          { nome: 'Autor 2', biografia: 'Nasceu em 1993, em Londres.' },
8        ]);
9      });
10 };
11
```


seeds > 02_seed_data.js > seed > seed > ...

```
1 < exports.seed = function(knex) {  
2   return knex('Livros')  
3     .del()  
4   .then(function() {  
5     return knex('Livros').insert([  
6       { titulo: 'Livro 1', genero: 'Terror' },  
7       { titulo: 'Livro 2', genero: 'Romance' },  
8     ]);  
9   });  
10  };
```